(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 144 421 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
13.01.2010 Bulletin 2010/02

(21) Application number: **08300234.5**

(22) Date of filing: **08.07.2008**

(51) Int Cl.:
H04L 29/08 *(2006.01)* ***G06F 17/30*** *(2006.01)*
H04L 29/06 *(2006.01)* ***H04L 12/22*** *(2006.01)*

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MT NL NO PL PT RO SE SI SK TR
Designated Extension States:
AL BA MK RS

(71) Applicant: **GEMPLUS**
13420 Gémenos (FR)

(72) Inventors:
- **Charbonnier, Emilien**
  83740 La Cadiere d'Azur (FR)
- **Galland, Antoine**
  13420 Gemenos (FR)
- **George, Patrcick**
  13600 La Ciotat (FR)

(54) Method for managing an access from a remote device to data accessible from a local device and corresponding system

(57) The invention relates to a method 30 for managing an access from a remote device to data and/or at least one resource accessible from a local device. The local device comprises a browser. The remote device hosts a server, as a remote server. The method comprises a remote server connecting step in which the browser sends to the remote server a request 38 for loading data.

According to the invention, the remote server sends 310, through the browser, to a local server a request for connecting a local server, as response to the request for loading data, and the local server sends 316 data to the remote server, the local server being connected from the remote server to data storing means and/or at least one resource accessible from the local device.

The invention relates also to a corresponding system comprising a token and a terminal coupled with the token.

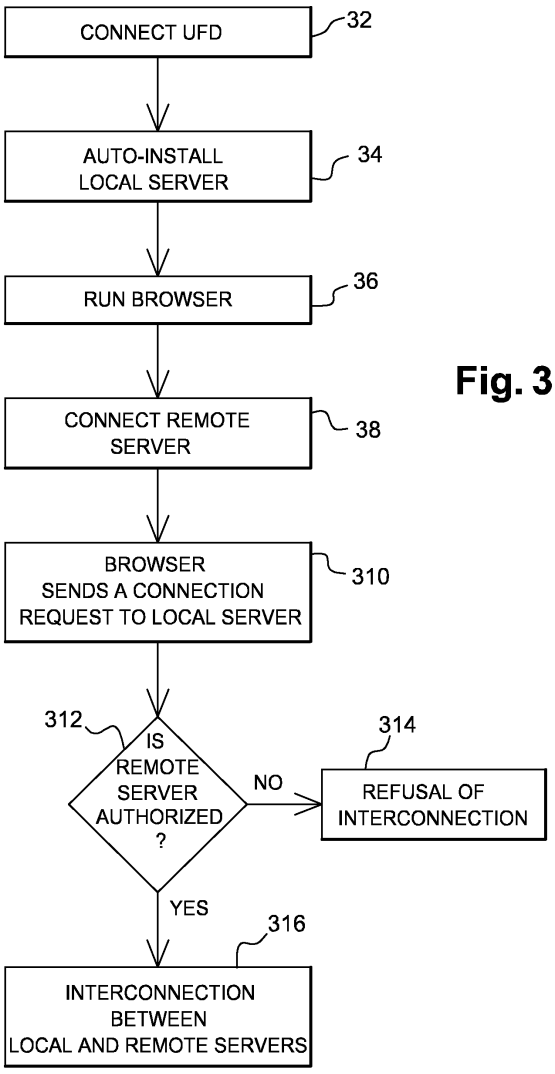

Fig. 3

EP 2 144 421 A1

Description

Field of the invention:

[0001] The invention relates, in a general manner, to a method for managing an access from a remote device to data and/or at least one resource accessible from a local device.

[0002] Moreover, the invention relates to a system for managing an access from a remote device to data and/or at least one resource accessible from a local device.

[0003] More particularly, the present invention applies to a system including a personal computer (or PC), as a local device, and a remote device connected to the local device.

State of the art:

[0004] The local device and the remote device are typically connected over a communication network, such as an Internet network or an Intranet network. The remote device hosts a server, termed, within the present description, a remote server.

[0005] As known per se, a PC user accesses the remote server by using one user interface, such as a (web) browser. Furthermore, the PC user accesses data stored locally, i.e. within the PC, or any hardware means connected to the PC, by using another user interface.

[0006] However, the local device user needs to use two different user interfaces, so as to access data stored within different storage areas. More exactly, one user interface is used for a remote access to data stored within a remote device, and another user interface is used for a local access to data and any local resource accessible from the local device.

[0007] Consequently, the local device user has to toggle between the two user interfaces, so as to perform some operations of cross-connection between the different data storage areas while being unable to perform some cross-connection operations.

[0008] A first known solution proposes to configure, in a complex manner, a (web) browser, so as to be able to perform some cross-connection operations between the different data storage areas.

[0009] Nevertheless, a "normal user" is not able to perform such a complex configuration that requires to be rather an "experience user".

[0010] A second known solution proposes to install a dedicated software component on the local device of the user, so as to be able to perform some cross-connection operations between the different data storage areas.

[0011] However, such known solutions have a major disadvantage.

[0012] As a matter of fact, some cross-connection operations are not possible. In particular, since the remote server does not access to data locally stored and/or any resource connected to the local device, a transfer of data from a local data storage area to a remote data storage area and a transfer of data from a remote data storage area to a local data storage are impossible.

Summary of the invention:

[0013] The invention eliminates such a major disadvantage by providing a method for managing an access from a remote device to data and/or at least one resource accessible from a local device. The local device comprises a browser. The remote device hosts a server, as a remote server. The method comprises a remote server connecting step in which the browser sends to the remote server a request for loading data.

[0014] According to the invention, the method comprises, as steps, a local server interconnecting step in which the remote server sends, through the browser, to a local server a request for connecting a local server, as response to the request for loading data; and a local server connecting step in which the local server sends data to the remote server, the local server being connected from the remote server to data storing means and/or at least one resource accessible from the local device.

[0015] The principle of the invention consists in addressing, through a (web) browser, a local server that interfaces between the remote server and the local data and/or hardware accessible from the local device.

[0016] The local server acts as a router between a local resource and the remote server.

[0017] The local server is able to transfer data from the remote server to any resource comprised within the local device and/or any resource connected to the local device.

[0018] The local server is able to transfer from any resource comprised within the local device and/or any resource connected to the local device to the remote server.

[0019] Thus, the remote server has to be connected to the local server, in order to exchange data with the local device resources or its peripheral resources.

[0020] The local server is therefore involved to let an access from the remote server to data stored locally or any resource connected to the local device. Likewise, the local server is also involved to access the remote server itself or through the remote server to another remote server to which the remote server allows an access.

[0021] The proposed solution allows therefore cross-connection operations for a user between the remote server and the involved local server.

[0022] Due to the fact that the remote server has to exchange with the local server, as an intermediary with any data stored locally or any hardware locally accessible, there is no need for a local device user to configure in a complex manner the browser. Moreover, there is either no need to install a dedicated software component on the local device.

[0023] A user addresses a unique user interface, namely a browser, to access data that are locally stored and data that are remotely stored, as well as to perform cross-connection operations between the local resource

and the remote resource.

[0024] According to a further aspect, the invention is a system for managing an access from a remote device to data and/or at least one resource accessible from a local device. The system comprises a local device and a remote device connected to the local device. The remote device hosts a server, as a remote server. The local device comprises a browser comprising means for sending to the remote server a request for loading data.

[0025] According to the invention, the local device comprises a local server; the remote server comprises means for sending, through the browser, to the local server a request for connecting a local server, as response to the request for loading data; the local server comprises means for sending data to the remote server, and the local server being configured so as to be connected from the remote server to data storing means and/or at least one resource accessible from the local device.

[0026] As local device, it can be, for example, a terminal, a desktop computer, a laptop computer, a handset, a mobile telephone, and/or a Personal Digital Assistant (or PDA).

Brief description of the drawings:

[0027] Additional features and advantages of the invention will be more clearly understandable after reading a detailed description of two preferred embodiments of the invention, given as indicative and not limitative examples, in conjunction with the following drawings:

- Figure 1 illustrates a simplified diagram of an embodiment of an electronic system comprising a PC, as a local device, and a remote device hosting a remote server, and being adapted to involve, through a browser, a local server between the remote server and data and/or any resource locally accessible according to the invention;
- Figure 2 illustrates a simplified diagram of an embodiment of a USB Flash drive, as token, to be connected to the PC of figure 1;
- Figure 3 is a flow chart of an exemplifying method for managing an access from the remote device to data and/or any resource locally accessible from the token of figure 2;
- Figure 4 depicts a first embodiment of a message flow involving the local server between the remote server and a local data storage area for a remote back-up of data locally stored, as the different entities implementing the method of figure 3;
- Figure 5 depicts a second embodiment of a message flow involving the local server between the remote server and a local data storage area and the USB Flash Drive of figure 2 connected to the PC of figure 1 for authenticating a local device user, as different entities implementing the method of figure 3 ;
- Figure 6 depicts a third embodiment of a message flow involving the local server, the remote server, an authentication server, and the USB Flash Drive of figure 2 for an online authentication of the local server, as different entities implementing the method of figure 3; and
- Figure 7 depicts a fourth embodiment of a message flow involving the local server, the remote server, an authentication server, a remote data storage area, for locally restoring data that is remotely stored, as different entities implementing the method of figure 3.

Detailed description of one embodiment:

[0028] Herein under is considered a case in which the invention method for managing an access from a remote device is implemented by a host computer, as a remote device embedding at least one server, to data and several peripheral devices accessible from a PC, as a local device.

[0029] However, it is only for exemplifying purposes and is not considered to reduce the scope of the present invention.

[0030] For example, instead of embedding at least one server, the remote device is connected to at least an external (remote) server. In such an example, the remote device becomes an intermediary entity between the exterior and the external server.

[0031] Figure 1 shows an electronic system 10 comprising a PC 12, as a local device, connected to at least one remote computer 18, as a remote device.

[0032] According to other embodiments, the local device is any terminal to which one or several devices can be connected, like a laptop computer, a handset, a mobile telephone, and/or a PDA.

[0033] As known per se (not represented), the PC 12 comprises, among others, one microprocessor, as data controlling and processing means, one hard disk, as mass-storage memory to store data relating to an Operating System, several applications including a (web) browser, and PC user data.

[0034] The PC 12 includes, as a man machine interface, a display screen 11 for displaying information to a PC user, and a keyboard 13 used by the PC user notably for entering and/or selecting data.

[0035] The PC 12 is connected, via a wireless link 15, like a Bluetooth (registered trademark) or a Wifi (registered trademark) link, to a communication network 16, such as Internet.

[0036] According to another embodiment, the PC 12 is linked through a wire link to Internet.

[0037] According to a variant, the local device is connected, through two communication networks, namely a mobile radio-telecommunication network, such a GSM (acronym for "Global System for Mobile"), and an Internet network, to the remote computer 18.

[0038] The PC 12 cooperates with a token.

[0039] Within the present description, a token is an electronic portable smart object that includes means for

processing and controlling data, such as at least one microprocessor.

[0040] As token, it can be any electronic portable device comprising at least one microprocessor, at least one memory or being intended to be connected to one memory, and an Input/Output communication interface.

[0041] For example, it can be any form factor, a dongle of the USB (acronym for "Universal Serial Bus") type (that does not need any specific reader within a host computer), a smart card, a MMC (acronym for "MuItiMediaCard") card, or a SD (acronym for "Secure Digital") Card, or any other electronic support that may have different form factors. According to another example, the token can also be a chip fixed, in a removable manner, to a host computer, or a chip to be soldered within a host computer.

[0042] A USB Flash Drive or UFD 14, as token, is coupled to the PC 12, as host and local device.

[0043] The UFD 14 is connected, through a USB connector, to a USB port of the PC 12.

[0044] According to a variant, the token is connected to the host device, through a wireless link, like a Bluetooth or a Wifi link.

[0045] The UFD 14 represents a peripheral device with respect to the PC 12, as its host computer.

[0046] The UFD 14 is described more in details in relation with figure 2.

[0047] The UFD 14 stores, within a client agent, a software component constituted by a local web server (not represented) termed, for simplicity, hereinafter local server.

[0048] Once connected, the UFD 14 is preferably configured to install and let run the local server on the PC 12.

[0049] The UFD 14 can store one or several additional software components, like one or several plug-ins, to be installed on the PC 12. The plug-ins interact with the web browser, as a host application, to provide one or several specific functions "on-demand", in order to create capabilities to extend the web browser. For example, the capabilities are rights for reading and/or writing a file(s), in order to transfer one or several files that can be edited with the installed plug-in.

[0050] The UFD 14 may also run the local server on its own, i.e. on the UFD 14.

[0051] The remote computer 18 is preferably connected to Internet 16 over a wire link 17.

[0052] The remote computer 18 can be connected to Internet 16 over a wireless link.

[0053] The remote computer 18 hosts a remote web server 110 termed, for simplicity, the remote server 110, a remote authentication server 112 and a remote (data) storage area 114.

[0054] Instead of also hosting the remote authentication server and the remote storage area, the remote computer 18 can be connected to one or two other remote computers hosting the remote authentication server and the remote storage area.

[0055] The remote server 110, the remote authentication server 112 and/or the remote storage area 114 can be merged and thus constitute one and the same server.

[0056] The remote server 110, the remote authentication server 112 and the remote storage area 114 can access data relating to a remote computer domain. The remote computer domain includes all the data and resources that are accessed and controlled by or through the remote server 110. Such a remote computer domain is considered as being a trusted domain. Among the data that is controlled, there is data relating to encryption and/or decryption, also termed credentials that are used to exchange data between the local server, the remote server 110, the remote authentication server 112 and/or the remote storage area 114.

[0057] The remote authentication server 112 is dedicated to generate authentication data or credentials to be used for an authentication of the local server before the remote server 110 and/or the remote storage area 114. Thus, the remote authentication server 112 ensures that no other server than the local server is allowed to communicate with the remote server 110 or the remote storage area 114.

[0058] The remote storage area 114 can be used as an online remote data storage area.

[0059] The remote storage area 114 can be used as a back-up of data already stored within the PC 12 or within any hardware resource connected to the PC 12, like the UFD 14.

[0060] The remote computer 18 can be connected to one or several other remote computers (not represented) that include corresponding remote storage areas.

[0061] According to a variant, instead of hosting the remote storage area 114, the remote computer 18 is connected to another remote computer (not represented) which incorporates a remote storage area.

[0062] To access the remote web server 110, the (web) browser runs and the PC user either types an Internet Protocol (or IP) address relating to the remote server 110 or selects its IP address within a predefined list, in order to send a request for loading data relating to a home page to be downloaded from the remote server 110.

[0063] According to the invention, when the remote server 110 responses to the request for loading data relating to a web page, the remote server 110 incorporates other data to be executed by the browser in order to address a local server to be involved, in order to exchange with the remote server 110.

[0064] According to a preferred embodiment, the UFD 14 is adapted to associate the local server and the remote web server 110 to be interconnected.

[0065] According to a variant, instead of being provided by a token, the local server is previously downloaded from a predetermined remote server, like the remote server 110, to the PC 12.

[0066] The local server is preferably configured to be addressed by at least one remote server 110 that is predetermined.

[0067] The local server has a predetermined IP ad-

dress that identifies it, in a unique manner.

[0068] For example, the IP address of the local server is the following "127.0.0.1:3516", in which a first part, namely "127.0.0.1", is used to access the PC 12, as its IP address, and a second part, namely ":3516", is used to address a particular port dedicated to the local server. This IP address can be defined by an organization, such as the Internet Assigned Numbers Authority (or IANA).

[0069] The local server is an application with user rights. The local server therefore accesses local file systems, namely data and/or resource(s) locally accessible, i.e. from the PC 12.

[0070] The local server ensures the cross-connection between outside and the resource(s) that is(are) locally accessible.

[0071] Thanks to the invention, the local server interfaces to access from the remote server 110 to data stored within local data storage areas and the peripheral hardware devices locally accessible. The local server therefore becomes a mandatory interlocutory from the remote server 110.

[0072] Then, the local server transfers data to the remote server 110.

[0073] Once interconnected, the local server and the remote server 110 can communicate to upload data (i.e. from the local server to the remote server 110 or a server connected to the remote server) and/or download data (i.e. from the remote server 110 or a server connected to the remote server to the local server).

[0074] The local server allows to access, in a controlled manner, from the remote server 110, the local data storage area(s) and resources accessible from the local device.

[0075] The local server preferably allows to access, in a controlled and secured manner, from the remote server 110, to the local data storage area(s) and resources accessible from the local device.

[0076] The local server allows to process data stored within local data storage area(s) and data stored within remote data storage area(s).

[0077] The local server is preferably so configured so as to be connected from the remote server 110 to data and one or several resources accessible from the PC 12. As resources accessible from the PC 12, it is in particular any peripheral device connected to the PC 12.

[0078] According to another embodiment, the local server is configured so as to be connected from the remote server only to data stored within the memory(ies) of the PC, namely its hard disk.

[0079] According to another embodiment, the local server is configured so as to be connected from the remote server only to a selection of peripheral devices connected to the PC 12, for example only the UFD 14, as one peripheral device.

[0080] According to a variant, the local server is configured so as to be connected from the remote server 110 to data and a selection of resources accessible from the PC 12.

[0081] As peripheral devices, besides the UFD 14, a printer 19 is connected to the PC 12.

[0082] Other peripheral devices can be also connected to the PC 12, in order to expand the capabilities of the PC 12. As peripheral devices, among others, it can be a scanner, an external disk drive, a tape drive, a microphone, a speaker, a camera, an external computer, and/or a card reader.

[0083] All the peripheral devices are controlled from the PC 12 through the local server.

[0084] The remote server 110 contains one or several web pages.

[0085] A web page is a document written in a language such as a HyperText Markup Language or HTML.

[0086] The web page is accessed via a HyperText Transfer Protocol (or "HTTP") using a schema based upon request/response messages, between a client (initiating a request message sent to the server) and a server (responding to the client with a response message).

[0087] To access the web page, the PC user runs the web browser and either types a Web address (or termed "Uniform Resource Locator") associated with the web page (also termed home page) of the remote web server 110 or selects the Web address associated with the remote server 110.

[0088] According to one aspect of the invention, the remote server 110 is adapted to send, through the browser, to the local server, as response to the request for loading data, a request for connecting the local server, to interconnect the remote server 110 and the local server.

[0089] The PC user sees, at the display screen 11, through the (web) browser, the desired web page transferred from the remote web server 110.

[0090] To interconnect the remote server 110 and the local server, the remote server 110 addresses to the browser, within the downloaded web page, an interconnecting message, as response to the request for loading data.

[0091] The downloaded web page transports executable data that is meant for the local server while using the predetermined IP address identifying the local server.

[0092] The web browser interprets and executes executable data, such as a JavaScript included within the downloaded web page, requesting it to address the local server. As known per se, JavaScript is a scripting language that has been standardized by Ecma (stands for European Computer Manufacturers Association) International within the ECMA-262 specification.

[0093] The web browser detects the presence of the local server and addresses the local server by implementing HTTP.

[0094] The downloaded web page contains a graphic interface that aggregates data originating from local data storage(s) and from remote data storage area(s) due to the local server and the remote server respectively.

[0095] The local server and the remote server 110 dynamically cooperate with each other to allow to transfer

data, in both directions between the different data storage areas, namely from local storage area(s) to remote storage area(s) and from remote storage area(s) to local storage area(s).

[0096] The invention application is constituted by the local server and the remote server 110. The invention application is therefore distributed between the PC 12, as the local computer, and the remote computer 18.

[0097] Prior to any further communication between the local server and the remote server, the remote server 110 is preferably to be authenticated by the local server.

[0098] To authenticate the remote server 110, the remote server 110 sends, through the browser, to the local server, as response to the request for loading data, data relating to its own credentials.

[0099] Prior to any further communication between the local server and the remote server, a user authentication is preferably implemented. Only a PC user desiring to connect, through the local server, to the remote server 110 that has its own credentials, is allowed to connect the remote server 110. No third party other than the authenticated PC user is allowed to connect the remote server 110.

[0100] To authenticate a user desiring to connect, through the local server, to the remote server 110, a PC user can enter, after having been prompted by a message displayed on the display screen 11 data, such as a Personal Identification Number (or PIN), as her or his credentials, by using the keyboard 13.

[0101] The message is displayed, thanks to executable data included within the downloaded web page originating from the remote server 110, for example, while using a script sent by the remote server 110 to the local server and executable by the local server, like a JavaScript, such as a "login.html".

[0102] The UFD 14 is preferably able to compare the entered data with an expected PIN stored within the UFD 14. The UFD 14 is able to authorize or forbid any further communication between the local server and the remote server 110. To inform the remote server 110 of a corresponding user authentication result, namely the user is authorized or not authorized, the UFD 14 sends, through the local server, to the remote server 110 a message comprising the corresponding user authentication result.

[0103] Instead of the UFD 14, a smart card (not represented) connected (through a contact reader) to the PC 12 is dedicated to compare the entered data with an expected PIN stored within the smart card. The smart card is able to authorize or forbid any further communication between the local server and the remote server 110. To inform the remote server 110 of a corresponding user authentication result, namely the user is authorized or not authorized, the smart card sends, through the local server, to the remote server 110 a message comprising the corresponding user authentication result. The local server and the smart card exchange, as network elements, by implementing, for example, a smart card protocol based on Application Protocol Data unit (or APDU).

[0104] The PC user that owns the UFD 14 and/or the smart card has the control for accessing (personal) data stored locally or remotely. When the PC user enters data that the smart card has to compare with the expected PIN, the PC user gives, as a voluntary action, her or his proof of consent for accessing (personal) data stored locally or remotely.

[0105] Alternatively, instead of PC user authentication, the local server and the remote server 110 can implement, or let implement, on its behalf, a mutual authentication.

[0106] To perform such a mutual authentication, the local server and the remote server 110 accesses one common Public Key Infrastructure (or PKI) for server authentication based on shared keys and a common encrypting/decrypting algorithm, like a DES (acronym for "Data Encryption Standard"), or a 3 DES, in order to exchange data in an encrypted manner.

[0107] As known per se, a PKI is a mechanism that enables different entities, as parties, to be authenticated to each other and to use the public key information to encrypt messages to each other. The PKI enables the parties in a dialogue to establish confidentiality, message integrity and user authentication without having to exchange any secret information in advance. The parties trust each other on their identities and keys.

[0108] When the PC user connects its UFD 14, the PC user gives her or his proof of consent by releasing credentials stored within the UFD 14 and accessible from the local server for its authentication before the remote server 110.

[0109] The local server accesses its own private key, called hereinafter local private key, and a public key relating to the remote server 110, called hereinafter remote public key that are stored within the UFD 14.

[0110] The UFD 14 implements the PKi by using the common encrypting algorithm and the remote public key, in order to encrypt data to be sent to the remote server 110.

[0111] The UFD 14 implements the PKi by using the common decrypting algorithm and the private local key, in order to decrypt data to be received from the remote server 110.

[0112] Likewise, the remote server 110 stores a private key relating to the remote server, called hereinafter the remote private key, a public key relating to the local server, called hereinafter the local public key, and one common encrypting/decrypting algorithm, in order to exchange data with the local server in an encrypted manner.

[0113] The remote server 110 implements the PKi by using the common encrypting algorithm and the local public key, in order to encrypt data to be sent to the local server.

[0114] The remote server 110 implements the PKi by using the common decrypting algorithm and the remote private key, in order to decrypt data to be received from the local server.

[0115] Once the remote server 110 and the local server

have been interconnected and mutually authenticated, the PC user can request, through the web browser, any operation of data transfer.

[0116] The web browser, as one single user interface, aggregates data originating from different remote data storage areas, and data originating from different local data storage areas.

[0117] More precisely, the web browser presents information relating to, on the one hand, data stored within the remote server 110, within the remote authentication server 112 and within the remote storage area 114, and, on the other hand, data stored within the PC hard disk, data stored within the UFD 14, data relating to the printer 19.

[0118] The web browser allows the PC user to manage access from the remote server 110 to data and resource (s) that are present within the PC 12 or connected to the PC 12, as well as access from the local server to data and resource that are stored within or accessible to the remote server 110.

[0119] The local server and the local data storages, as local resources, namely the hard disk and the UFD 14, can exchange by using a file system.

[0120] As known per se, a file system is a method for storing and organizing (data) files and the data that the files contain to make it easy to retrieve and access the data.

[0121] In particular, the PC user can request to transfer data stored within the hard disk, as PC memories, or stored within the UFD 14, to the remote computer 18 or other remote computer connected to the remote computer 18 (not represented).

[0122] More precisely, the PC user can select several files stored locally, for example, within the UFD 14, in order to upload them to the remote data storage area 114.

[0123] To perform such an upload of selected files, a simple user operation, like "drag" selected files from one local data storage and "drop" the selected files to one remote local data storage, is required.

[0124] The PC user is able to access, on one and the same web page that can be seen through the web browser, data accessible from the remote server 110 and data accessible from the local server.

[0125] The PC user can request to access the printer 19, for example, to print a document comprising a page that presents all data that is locally and/or remotely stored with its storing location.

[0126] Once the PC user has performed the operation (s) she or he desires, the UFD 14 can be unplugged while removing, from the PC 12, on the one hand, traces of data relating to the performed operation(s), and, on the other hand, the data and/or peripheral resource(s) that is(are) locally accessible.

[0127] Figure 2 schematically shows components incorporated within, as token, the UFD 14 intended to be connected to a host computer, as a local device.

[0128] According to one important aspect, the UFD 14 is suited to associate a local server and an external server, as a remote server, to be interconnected. The UFD 14 is suited to configure the local server so as to be connected, through a browser, from the remote server to data storing means and/or at least one resource accessible from the local device.

[0129] In other words, the token allows to pair, through a browser, a local server with an external server, so that a device hosting the token, as the local device, is set to connect the external server to the local server, as an interlocutor of the external server.

[0130] Accordingly, contrary to the first and second aforementioned known solutions, the local device user has not any operation to perform to involve the local server, as the interlocutor of the remote server, for an access from outside to data stored within the local device and/or any resource accessible from the local device.

[0131] The proposed solution is convenient for the user and therefore user-friendly.

[0132] The UFD 14 comprises data controlling and processing means 22, such as one microprocessor, data storing means 24, and an Input/Output interface 26 connected together with a data and control bus 23.

[0133] The Input/Output interface 26 includes a USB connector. The USB connector is intended to be connected to a USB port connected to or provided by a host computer, such as the PC 12.

[0134] The UFD 14 can comprise, as a man machine interface, a display screen 28 for displaying information to a user and one button 210 for validating an operation. The man machine interface allows a real interaction between a UFD user and the UFD 14 itself in a connected mode or in an unconnected mode. The UFD user can use the man machine interface to perform One Time Password (or OTP) operations, in order to generate a key required to any operation, like:

- an access to the data stored locally;
- a transfer of data stored locally to a remote resource, such as a remote server, a remote authentication server, and/or a remote storage area;
- a transfer of data stored within a remote storage area to a local resource, such as a Hard Disk, as a PC memory, and/or the UFD 14.

[0135] The UFD 14 is able to cooperate with any host computer, such as the PC 12, to operate as a local server with respect to a remote server.

[0136] The data and processing means 22, as heart of the UFD 14, processes and controls data to be exchanged within the UFD 14 and also data to be exchanged with outside the UFD 14.

[0137] The data storing means 24 preferentially includes volatile and non volatile memories. The non volatile memory can be constituted by one or several EEPROM (acronym for "Electrically Erasable Programmable Read-Only Memory"), one or several ROM (acronym for "Read Only Memory"), one or several Flash memories, and/or any other memories of different types, like one or

several RAM (acronym for "Random Access Memory").

[0138] The data storing means 24 stores an Operating System for operating the UFD 14, and one or several applications.

[0139] The data storing means 24 comprises a mass-storage area for storing personal user data, such as photos, music and/or video data.

[0140] The data storing means 24 preferably comprises credentials, such as a PIN and/or keys, among which an encrypting key, a decrypting key. The credentials can be used to securely store data within the UFD 14 and/or outside the UFD 14, authenticate the user and/or the local server, sign data to be sent to outside world.

[0141] The data storing means 24 preferably stores a local server.

[0142] The data storing means 24 stores at least one identifier for each remote server that is allowed to interconnect the local server.

[0143] The data storing means 24 can store at least one function library to provide a host web browser with some extension capabilities, such as an access for reading a particular type of data file. Such an access for reading a particular type of data file allows a UFD user to edit a data file before requesting to transfer it.

[0144] The local server can be run on the UFD 14 itself.

[0145] The data storing means 24 preferably comprises at least one working memory that is accessible in particular when the local server is being executed.

[0146] The local server is preferably to be run on a host computer.

[0147] The local server is configured so that it can only be addressed by one or several predetermined remote servers that are identified and also preferably authenticated by the local server.

[0148] Furthermore, the local server is configured so that, when running, the local server accesses at least one local memory, i.e. one or several memories comprised within the host computer or connected to the host computer, and, in particular, the mass-storage area of the UFD 14.

[0149] The personal user data stored within the mass-storage area can be uploaded, at least in part, through the local server that runs, by transferring data to the remote storage area 114.

[0150] The personal user data can be downloaded, at least in part, onto the mass-storage area, from data originating from the remote storage area 114 through the local server that runs.

[0151] According to a preferred embodiment, once the microprocessor 22 has been informed that the USB connector 26 is connected to a host computer, the operating system is adapted such that the local server is automatically launched (either on the host computer or on the UFD) while configuring an access to the local server and an access from the local server to any hardware comprised or connected to the host computer.

[0152] Such an auto-run feature reduces the UFD user intervention allowing any UFD user to use it.

[0153] The data storing means 24 stores at least one application providing at least one security function, such as a secure data storage, a digital signature, a user identification, a user authentication, an on-board key generation, and/or a secure exchange for online transactions.

[0154] The invention token allows to obtain a portable electronic support that, besides a portable mass-storage area, provides a user-friendly (portable) token that interposes by itself a local server to communicate with an external server.

[0155] The token user does not need to intervene, in order to possibly install the local server and configure an interconnection between the local server and the remote server.

[0156] The token user has just to couple the token with a host computer. Then, the token user manipulates, in an instinctive manner, for example, one or several graphic elements displayed on a single user interface, namely a web browser.

[0157] The token user can perform different operations through the web browser, such as to read, write one data file stored either locally or remotely, download, or upload one or several data files.

[0158] The token user is able to transfer, through the local server, data to be stored within a remote storage area for an online remote back-up or data originating from a remote data storage to be stored on the mass-storage area of the token.

[0159] Figure 3 depicts one example of a method for managing an access from the remote device 18 to data and/or any resource locally accessible from the UFD 14.

[0160] Firstly, an owner of the UFD 14 connects 32 the UFD to the PC 12, as the local device.

[0161] Once the UFD is connected, the UFD auto-installs 34 on the PC the local server stored on the UFD. The local server is configured so as to connect to the local resources, namely the PC memories, the UFD and any other peripheral device connected to the PC, like a printer.

[0162] A PC user launches 36 a web browser by using the PC man machine interface.

[0163] The PC user connects 38, through the web browser, to a remote server by sending a request for downloading data relating to its home page. To connect to the remote server, the PC user either types the home page address or selects the remote server name within a list of at least one predefined server name.

[0164] Then, the web browser sends 310 to the local server, a request for connecting a local server. To send the request for connecting the local server, the remote server sends, for example, by using a script, data executable by the PC web browser, downloaded within the data relating to the home page, as a response to the request for downloading data relating. The web browser interprets and executes the script by directing the request for connecting a local server to the local server installed within the PC.

[0165] The PC runs the local server that is inserted as

an interlocutor.

[0166] The local server verifies 312 whether the remote server is identified as an allowed remote server to be connected to the local server. To verify whether the remote server is authorized by the local server, the local server compares the remote server identifier with an identifier of each allowed remote server comprised within an identifier list. The identifier list is stored within the UFD.

[0167] Optionally, the local server verifies at least one other conditional access is fulfilled to authorize the remote server to access the local resource(s).

[0168] According to a variant, once the local server has received, through the browser, the request for connecting the local server, the local server does not verify any conditional access, the local server sends data to the remote server. No verification step is required.

[0169] When the local server does not identify the remote server trying to interconnect with the local server, the local server refuses 314 an interconnection with the remote server.

[0170] The local server can send a message for informing the remote server that it is not allowed to access any local resources. The local server forbids any access to local resource, such as data files stored within the PC and/or the UFD.

[0171] On the contrary, i.e. when the remote server is identified by the local server, as having an authorization to connect to the local server, then the local server accepts 316 an interconnection with the remote server.

[0172] Thanks to the web browser, the local server becomes a bridge between any connected local resource and the interconnected remote server, to access either the connected local resource from the interconnected remote server or any connected remote resource from the local server.

[0173] Figure 4 shows messages involving the local server between the remote server and a local data storage area for a remote back-up of data stored within the PC.

[0174] Different entities, namely a Hard Disk 42, a (web) browser 44, a remote (web) server 110 (or remote web site) and a local (web) server 48 exchange 40 data for a transfer of one selected (data) file stored within a memory of the PC, as local resource, to a remote data storage.

[0175] A vertical dotted line 46 separates the local resource(s) on the left side from the remote resource(s) on the right side.

[0176] Firstly, the browser 44 sends to the remote web site 110 a request 410, such as an HTTP GET, for downloading data relating to a home page stored remotely.

[0177] The remote site 110 receives the request and sends a response 412, to the browser 44, comprising the home page including a command for involving a local server 48 and a list of accessible remote data files preferably accompanied by credentials pertaining to the remote server.

[0178] The remote server credentials are constituted by data relating to the remote data files and that has been encrypted with an encrypting key and an encrypting algorithm shared between the remote server 110 and the local server 48.

[0179] The browser 44 shows, through the downloaded home page, the accessible remote data files.

[0180] The browser extracts from the downloaded home page the command for involving a local server 48 and addresses a request 414, such as an HTTP GET for an XML (acronym for "eXtensible Markup Language") document, to the local server identified within the command while transmitting the remote server credentials.

[0181] The local server 48 checks whether the sender identifier is known to it and whether the remote server credentials are known to it. The local server 48 decrypts encrypted data received through the browser 44 by using a decrypting key and a decrypting algorithm common to the remote server 110 and the local server 48.

[0182] When applicable, i.e. when the remote server 110 is identified and the received remote server credentials are the ones stored, then the local server 48 sends to the hard disk 42 a request 416, while using an appropriate file system command, for giving data relating to data files that are accessible.

[0183] The hard disk 42 transmits to the local server 48 a message 418 with a file list that can be uploaded, as a local file list.

[0184] The local server 48 sends to the browser 44 a message 420, such a HTTP response, for displaying the local file list stored by the hard disk 42.

[0185] Once the browser 44 has received the message 420 and interpreted the file list to be displayed, the browser 44 displays, besides the remote data files, the local file list.

[0186] A PC user, through the PC man machine interface, can request an operation through the browser 44.

[0187] The browser 44 interprets an action requested by the PC user constituted by a desire for uploading several data files, as a selection of data files of the local file list.

[0188] The browser 44 sends to the local server 48 a request 422, such as HTTP GET selected local files, specifying the local files to be copied onto the remote server 110.

[0189] The local server 48 transfers the request by sending a request 424, such as a file system command, for transmitting a copy of the selected local files to the local server 48.

[0190] Then, the local server 48 accesses to the hard disk memory through a message 426 comprising data relating to the selected local files.

[0191] Once the local server 48 receives data relating to the selected data files, the local server 48 adds its own credentials to this data, for example by encrypting it with an encrypting key and an encrypting algorithm known to both the local server 48 and the remote server 110.

[0192] Then, the local server 48 sends a HTTP message 428 including the resulting encrypted data, as local

server credentials, to the remote server 110, in order to store the data sent by the hard disk 42. It is to be noted that the local server 48 transmits the HTTP message 428 to the remote server 428 in a direct manner, i.e. in a transparent manner with respect to the browser.

[0193] The remote server 110 receives the HTTP message 428 and checks the local server credentials, for example by decrypting the received data while using the common decrypting key and decrypting algorithm before storing the received data, as a back-up of the data stored within the hard disk 42.

[0194] Optionally, the remote server 110 accesses a remote (data) storage area within which the data relating to the selected data files is stored in an encrypted manner.

[0195] Figure 5 shows messages involving the local server between the remote server and a local data storage area for a remote back-up of data stored within the PC.

[0196] Different entities, namely a Hard Disk 42, a (web) browser 44, a remote (web) server 110 (or remote web site), a local (web) server 48 and a UFD 52, exchange 50 data for a transfer of one selected (data) file stored within a memory of the PC, as local resource, to a remote data storage.

[0197] Only the differences with respect to the description of figure 4 are herein under explained.

[0198] Once the local server 48 has received the message 414 accompanied by the remote server credentials, instead of verifying the remote server credentials by itself, the local server 48 sends to a UFD 14 (connected to the PC) a message 52 for requesting the UFD to verify the remote server credentials.

[0199] According to a variant, the local server 48 sends to the UFD 14 a message for requesting a decrypting key and a decrypting algorithm to be used to decrypt data originating from the remote server 110. The local server 48 is supplied with information to decrypt data originating from the remote server 110.

[0200] The UFD 14 verifies whether the remote server credentials are the ones stored within the UFD 14 for a remote server that is associated.

[0201] When the remote server credentials match with the ones stored within the UFD 14, then the UFD 14 sends to the local server 48 a message 54 for informing the local server 48 that the remote server is one that is authorized to communicate with the local server 48.

[0202] Once the local server 48 has received the message 426 comprising data relating to the selected local files, instead of adding the local server credentials by itself, the local server 48 delegates it to the UFD 14. To add the local server credentials, the local server 48 sends to the UFD 14 a message 56 for requesting the UFD to add the local server credentials accompanied by data relating to the selected local files.

[0203] According to a variant, the local server 48 sends to the UFD 14 a message for requesting an encrypting key and an encrypting algorithm to be used to encrypt data relating to the selected local files to be transferred to the remote server. The local server 48 is supplied with information to encrypt data relating to the selected local files to be transferred to the remote server 110.

[0204] The UFD 14 adds the local server credentials for example by encrypting the data relating to the selected local files with an encrypting key and an encrypting algorithm known to the remote server 110.

[0205] Then, the UFD 14 sends a message 58 including the resulting encrypted data while containing local server credentials.

[0206] Figure 6 shows messages involving the local server 48, the remote server 110, an authentication server 112, and the USB Flash Drive 14 for an online authentication.

[0207] According to another embodiment (not represented), the remote server and the authentication server constitute a single server.

[0208] These different entities exchange 60 data for a transfer of a user authentication key from the authentication server 112 to the local server 48.

[0209] The online authentication is performed by the authentication server 112.

[0210] Firstly, the browser 44 sends to the remote web site 110 a request 64, such as an HTTP GET, for downloading data relating to a home page stored within the remote server 110.

[0211] The remote server 110 receives the request 64. The remote server 110 sends a response 66, to the browser 44, encompassing a command for involving a local server 48 accompanied by credentials pertaining to the remote server.

[0212] The command for involving the local server 48, such as "login.html", is a script that is executable by the browser 44. Such executable data requires the browser 44 to involve the local server 48 to authenticate itself by providing credentials relating to the UFD 14.

[0213] The remote server credentials are constituted by a first secret data that has been encrypted with a private remote server key and an encrypting algorithm stored within (or accessible by) the remote server 110.

[0214] The first secret data and the corresponding public remote server key and decrypting algorithm are stored within (or accessible by) the local server 48.

[0215] The browser 44 extracts from the encrypted data, the command for involving a local server 48.

[0216] The browser then addresses a request 68, such as an HTTP GET, to the local server identified within the command while transmitting it the encrypted data, as remote server credentials, originating from the remote server 110.

[0217] The local server 48 checks whether the sender identifier is known to it, and preferably whether the remote server credentials are known to it. The local server 48 decrypts the encrypted data received through the browser 44 by using a public remote server key and a decrypting algorithm stored within the local server 48.

[0218] When applicable, i.e. when the remote server

110 is identified and preferably the received remote server credentials are the ones stored, namely the ones that allow to retrieve the first secret data by using the corresponding public remote server key and decrypting algorithm, then the local server 48 has authenticated the remote server.

[0219] The local server 48 sends to the UFD 14 a message 610 for requesting credentials relating to a local server, as local server credentials.

[0220] The local server credentials are constituted by a second secret data that has been encrypted with a private local server key and an encrypting algorithm stored within (or accessible by) the UFD 14.

[0221] The second secret data can be identical or distinct from the first secret data.

[0222] This second secret data and the corresponding public local server key and decrypting algorithm are stored within (or accessible by) the authentication server 112.

[0223] The UFD 14 sends to the local server 48 a message 612 comprising the local server credentials.

[0224] Once this latter message 612 has been received by the local server 18, the local server 48 forwards to the authentication server 112, thanks to another message 614, the local server credentials. With this same message 614, the local server 48 requests to the authentication server, a user authentication key, as a user authentication ticket, for the current communication session.

[0225] The user authentication ticket is to be shared between the local server and the remote server 110. The user authentication ticket is used to allow to subsequently exchange further data originating from either the local data storage area(s) or the remote data storage area(s) between the local server 48 and the remote server 112. More exactly, the user authentication ticket can be used by the local server 48 to encrypt data to be sent to the remote server 110.

[0226] After the authentication server 112 has received the message 614, the authentication server 112 uses the public local server key and the decrypting algorithm, in order to decrypt and retrieve the data sent by the local server 48.

[0227] Once the data has been decrypted, the authentication server 112 checks whether the decrypted data matches with the second secret data that is stored within (or accessible by) the authentication server 112.

[0228] On the contrary, when the comparison result is negative, then authentication server 112 does not authenticate the local server 48.

[0229] In such a case, the authentication server 112 does not send to the local server 48 any further message allowing the local server 48 to continue exchanging data with the remote server 48.

[0230] When the comparison is positive, namely the decrypted data matches with the second secret data, then the authentication server 112 authenticates the local server 48.

[0231] In such a latter case, the authentication server 112 sends to the local server 48 a message 616 containing a user authentication ticket.

[0232] The authentication server 112 can revoke the user authentication ticket possibly by sending a message to the local server for informing it about a revocation relating to the user authentication ticket.

[0233] Once supplied with the user authentication ticket, the local server 48 sends to the browser 44 a message 618 including the user authentication ticket. The message 618 is intended for the remote server 110.

[0234] The browser 44 receives the user authentication ticket.

[0235] Then the browser 44 forwards to the remote server 110, thanks to another message 620, the user authentication ticket.

[0236] According to a variant, instead of transmitting the user authentication ticket through the browser 44 to the remote server 110, the local server 48 sends, in a direct manner, to the remote server 110 the user authentication ticket.

[0237] The local server 48 may thus use the user authentication ticket to further exchange with the remote server 110. The local server 48 can use the user authentication ticket to encrypt data to be transferred to the remote server 110.

[0238] Eventually, only the remote server 110 that is paired is effectively allowed to discuss with the local server 48 that may access to the local resources and data.

[0239] Figure 7 shows messages involving the local server 48, the remote server 110, an authentication server 112, a remote data storage area 114, and the hard disk 42 for locally restoring data that remotely is stored.

[0240] According to another embodiment (not represented), the remote server, the authentication server and the remote data storage area constitute a single server.

[0241] These different entities exchange 70 data for a transfer of one or several selected (data) files stored within a memory that is remotely accessible, as remote resource, to a memory of the PC, namely the hard disk 42, as local data storage area.

[0242] It is assumed herein under that the local server has previously been provided with a user authentication key, as a user authentication ticket, as a first user authentication level.

[0243] Firstly, the browser 44 sends to the remote server 110 a request 74, such as HTTP GET, for obtaining a home page stored within the remote server 110.

[0244] The remote server 110 receives the request 74 and sends, to the browser 44, a response 76 encompassing a command for involving a local server 48 accompanied by credentials pertaining to the remote server.

[0245] The command for involving the local server 48, such as "portail.html", is a script that is executable by the browser 44. Such executable data requires the browser 44 to involve the local server 48 to authenticate itself by providing the authentication server 112 with the user au-

thentication ticket, as local server credentials.

[0246] The remote server credentials are constituted by a first secret data that has been encrypted with a private remote server key and an encrypting algorithm stored within (or accessible by) the remote server 110.

[0247] The first secret data and the corresponding public remote server key and decrypting algorithm are stored within (or accessible by) the local server 48.

[0248] The browser 44 extracts from the encrypted data, the command for involving a local server 48.

[0249] The PC user requests, through the browser 44, by using the man machine interface, a list of data files that are remotely stored, i.e. within a remote data storage area 114.

[0250] The browser 44 then addresses a request 78, such as an HTTP GET, to the local server identified within the command while transmitting it the encrypted data, as remote server credentials, originating from the remote server 110.

[0251] The local server 48 checks whether the sender identifier is known to it, and preferably whether the remote server credentials are known to it. The local server 48 decrypts the encrypted data received through the browser 44 by using a public remote server key and a decrypting algorithm stored within the local server 48.

[0252] When applicable, i.e. when the remote server 110 is identified and the received remote server credentials are the ones stored, namely the ones that allow to retrieve the first secret data by using the corresponding public remote server key and decrypting algorithm, then the local server 48 has authenticated the remote server.

[0253] The local server 48 sends to the authentication server 112 a message 710 for requesting credentials relating to a local server, as local server credentials. This message 710 includes a request for accessing the list of data files stored within the remote data storage area 114 and the user authentication ticket.

[0254] The authentication server 112 verifies whether the received data matches with the user authentication ticket.

[0255] When the comparison between the received data and the user authentication ticket is negative, the authentication server 112 does not authenticate the local server and forbids any further communication with the local server 48.

[0256] When the comparison between the received data and the user authentication ticket is positive, the authentication server 112 authenticates the local server 48.

[0257] In such a latter case, the authentication server 112 sends to the local server 48 a message 712 comprising an operation authentication key, as an operation authentication ticket, in order that the user be allowed to perform any operation with respect to the remote data storage area 114.

[0258] Once received, the local server 48 uses the operation authentication ticket, as a second user authentication level, for encrypting any message to be addressed to the remote storage area 114.

[0259] The authentication server 112 can revoke the operation authentication ticket possibly by sending a message to the local server for informing it about a revocation relating to the current operation authentication ticket or transmitting another operation authentication ticket.

[0260] The local server 48 sends to the remote data storage area 114 a message 714 for requesting to be provided with the file list stored within it.

[0261] After having authenticated the local server 48, the remote data storage area 114 sends to the local server 48, as a response, a message 716 including the available file list.

[0262] Once this message 716 has been received, the local server 48 sends to the browser 44 the file list that is accessible from the remote data storage area 114.

[0263] The PC user selects, among the file list, through the browser 44, a file that she or he desires to store locally, by executing a corresponding operation.

[0264] Then, the browser 44 interprets the operation and requests to select the desired file by sending to the local server a message 720 for requesting to transfer the selected file.

[0265] The local server 48 receives this message 720 with the file to be selected. Then, the local server 48 transmits to the authentication server 112 a message 722 for requesting to transfer from the remote data storage area 114 to the local server 48 the selected file and the operation authentication ticket, as local server credentials.

[0266] The authentication server 112 verifies whether the data received from the local server 48 matches with the operation authentication ticket.

[0267] When the comparison between the received data and the operation authentication ticket is negative, the authentication server 112 does not authenticate the local server and forbids any further communication with the local server 48.

[0268] When the comparison between the received data and the operation authentication ticket is positive, the authentication server 112 authenticates the local server 48.

[0269] In such a latter case, the authentication server 112 sends to the local server 48 a message 724 comprising another operation authentication key, as an operation authentication ticket, in order that the user be allowed to perform any operation with respect to the remote data storage area 72.

[0270] Once received, the local server 48 uses the other operation authentication ticket, as a third user authentication level, for encrypting any message to be addressed to the remote storage area 72.

[0271] The local server 48 sends to the remote data storage area 114 a message 726 for requesting to be provided with the selected file stored within the remote data storage area 72.

[0272] After having authenticated the local server 48,

the remote data storage area 114 sends to the local server 48, as a response, a message 728 including the available selected file.

[0273] Once this message 728 has been received, the local server 48 sends to the hard disk 42 a message 730 containing the selected file (that has been supplied by the remote data storage area 72).

[0274] Finally, the hard disk 42 receives the selected file to be written, in order to copy the selected file that is remotely stored.

Claims

1. A method (30) for managing an access from a remote device (18) to data and/or at least one resource accessible from a local device (12), the local device comprising a browser, the remote device hosting a server (110), as a remote server, the method comprising a remote server connecting step in which the browser sends to the remote server a request (38) for loading data,
   characterized in that the method comprises:

   - a local server interconnecting step in which the remote server sends (310), through the browser, to a local server a request for connecting a local server, as response to the request for loading data; and
   - a local server connecting step in which the local server sends (316) data to the remote server, the local server being connected from the remote server to data storing means and/or at least one resource accessible from the local device.

2. Method according to claim 1, wherein the method comprises a remote server authentication step, the remote server authentication step consisting in that the local server interconnecting step consists in that the remote server sends, through the browser, to the local server, a request (412, 414) for connecting a local server accompanied by data relating to first credentials, as response to the request for loading data, and
   the local server compares data relating to the first credentials with first expected authentication data, and, according to a comparison result, the local server authorizes or forbids the remote server to further exchange data with the local server.

3. Method according to claim 1 or 2, wherein the browser presents, through the local server, at least one item of information relating to, on one hand, data stored within the remote device and/or within at least one other device connected to the remote device and, on the other hand, :

   - data stored within the local device;
   - data stored within at least one other device connected to the local device; and/or
   - data relating to at least one resource (19) accessible from the local device.

4. Method according to any of claims 1 to 3, wherein the method comprises a user authentication step, the user authentication step consisting in that the local server sends data (614) relating to second credentials to the remote server and/or an authentication server, the remote server and/or the authentication server compares data relating to second credentials with second expected authentication data, and, according to a comparison result, the remote server and/or the authentication server forbids or authorizes the local server to continue while transmitting third credentials (616) to the local server to be used for further communicating from the local server to the authentication server and/or the remote server.

5. Method according to claim 4, wherein the method comprises a local server authentication step, the local server authentication step consisting in that the remote server and/or the authentication server receives from the local server data relating to the third credentials, the remote server and/or the authentication server compares data relating to third credentials with third expected authentication data, and, according to a comparison result, the remote server and/or the authentication server forbids or authorizes the local server to continue while transmitting fourth credentials (712) to the local server to be used for further communicating from the local server to the authentication server and/or the remote server.

6. Method according to claim 4 or claim 5, wherein, a token (14) being connected to the local device, the token is able to generate data relating to the second credentials and/or store data relating to second credentials to be sent by the local server to the remote server and/or the authentication server.

7. Method according to claim 4 or 5, wherein a user enters data, as second credentials, by using a man machine interface (13) included within or connected to the local device, the second credentials having to be sent by the local server to the remote server and/or the authentication server.

8. Method according to claim 6 or 7, wherein the remote server and/or authentication server are able to generate the data relating to the first, the third and/or the fourth credentials, and/or store data relating to the first, the third and/or the fourth credentials to be sent by the remote server and/or authentication server to the local server and/or store the first, the third and/or

the fourth expected authentication data.

9. Method according to any of claims 6 to 8, wherein the remote server and/or the authentication server is able to revoke the third and/or the fourth credentials while transmitting a corresponding message to the local server and/or a token connected to the local device.

10. Method according to any of claims 1 to 9, wherein, a token being connected to the local device, the method comprises a local server installing step in which the token installs (34) the local server within the local device.

11. Method according to any of claims 1 to 10, wherein the local server sends (428) to the remote device and/or another remote data storage means connected to the remote device data stored within the local device and/or within a resource connected to the local device and/or a token connected to the local device.

12. Method according to any of claims 1 to 11, wherein the remote device sends (728) to the local server data stored within the remote device and/or within another device connected to the remote device.

13. Method according to claim 11 or 12, wherein the remote device and the local server exchange data in an encrypted manner.

14. A system (10) for managing an access from a remote device to data and/or at least one resource accessible from a local device, the system comprising a local device and a remote device connected to the local device, the remote device hosting a server, as a remote server, the local device comprising a browser comprising means for sending to the remote server a request for loading data,
characterized in that:

- the local device comprises a local server;
- the remote server comprises means for sending, through the browser, to the local server, a request for connecting a local server, as response to the request for loading data;
- the local server comprises means for sending data to the remote server, and
- the local server being configured so as to be connected from the remote server to data storing means and/or at least one resource accessible from the local device.

15. System according to 14, wherein the local device comprises at least one element comprised within the following group :

- a personal computer;
- a desktop computer;
- a terminal;
- a handset;
- a mobile telephone;
- a personal digital assistant; and/or
- a laptop computer.

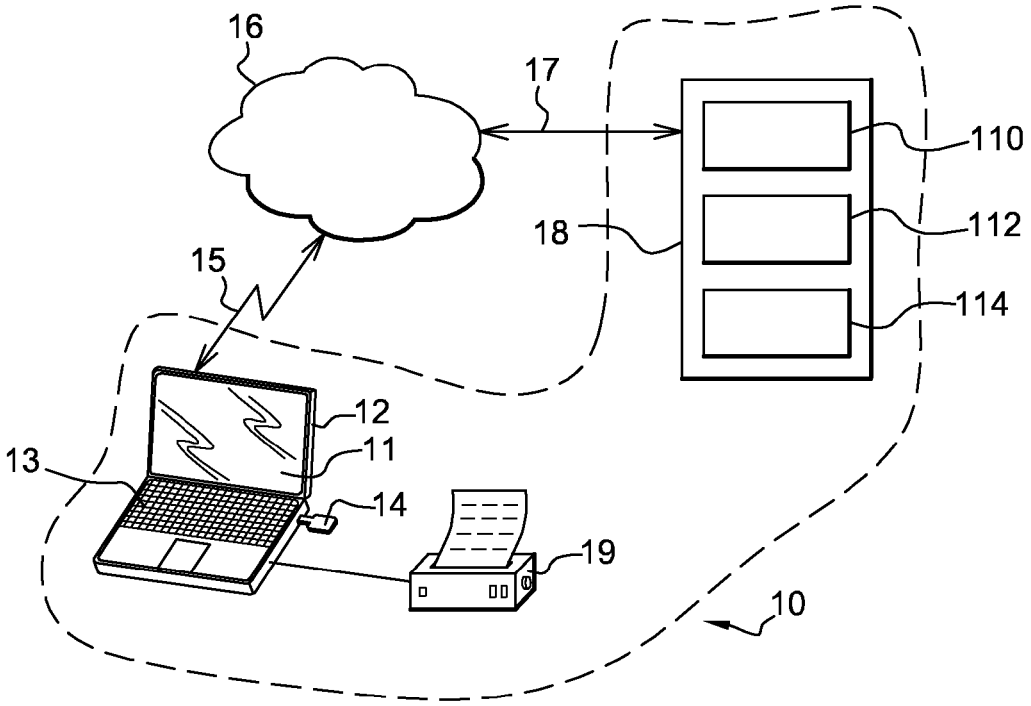

Fig. 1

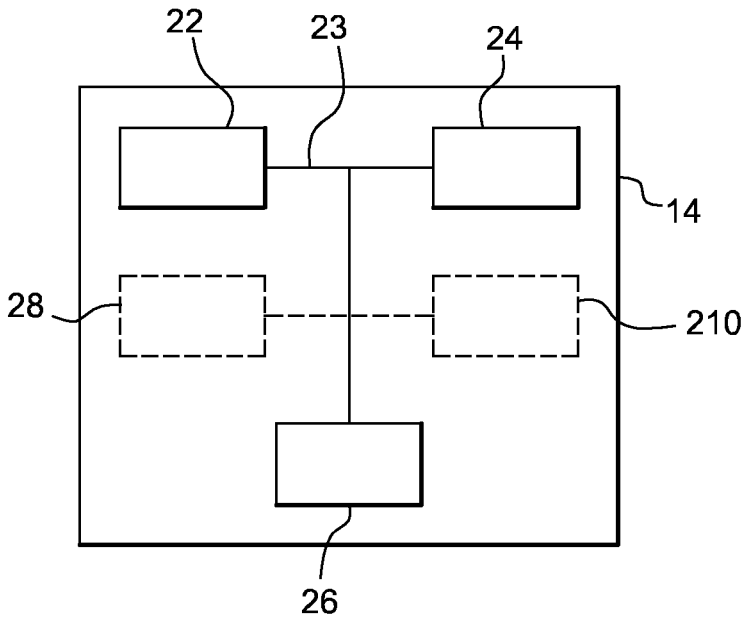

Fig. 2

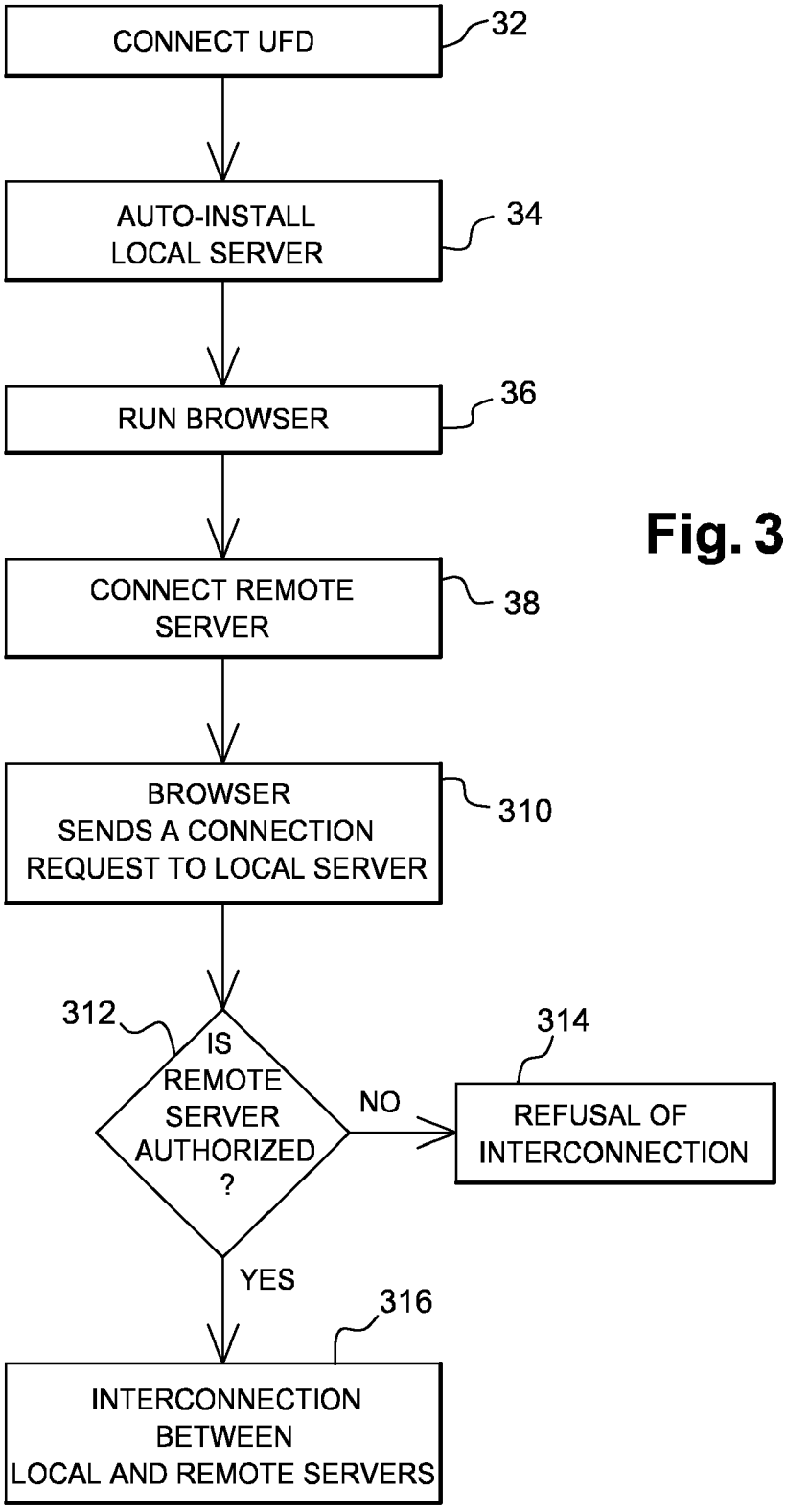
CONNECT UFD
32
AUTO-INSTALL LOCAL SERVER
34
RUN BROWSER
36
CONNECT REMOTE SERVER
38
BROWSER SENDS A CONNECTION REQUEST TO LOCAL SERVER
310
312
IS REMOTE SERVER AUTHORIZED ?
NO
314
REFUSAL OF INTERCONNECTION
YES
316
INTERCONNECTION BETWEEN LOCAL AND REMOTE SERVERS

Fig. 3

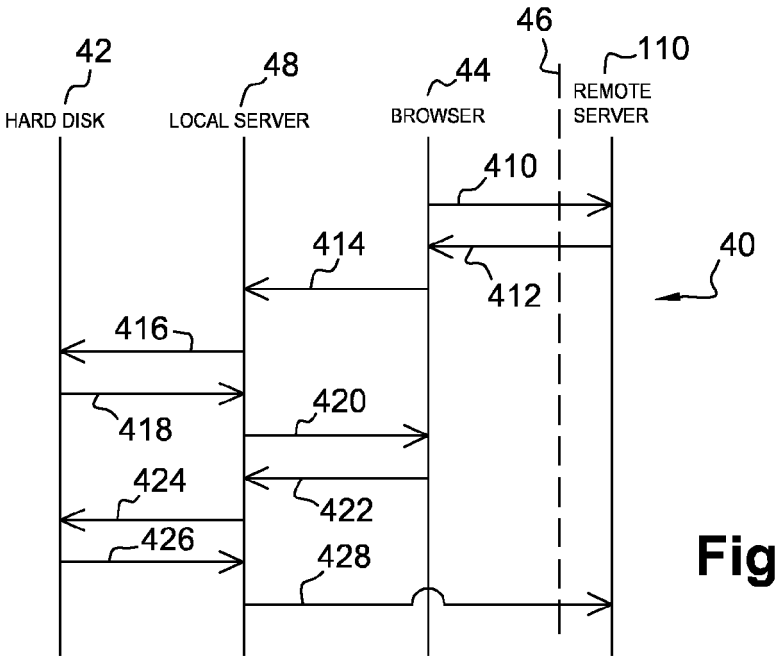
46
42
48
44
110
HARD DISK
LOCAL SERVER
BROWSER
REMOTE SERVER
410
414
40
412
416
418
420
424
422
426
428

Fig. 4

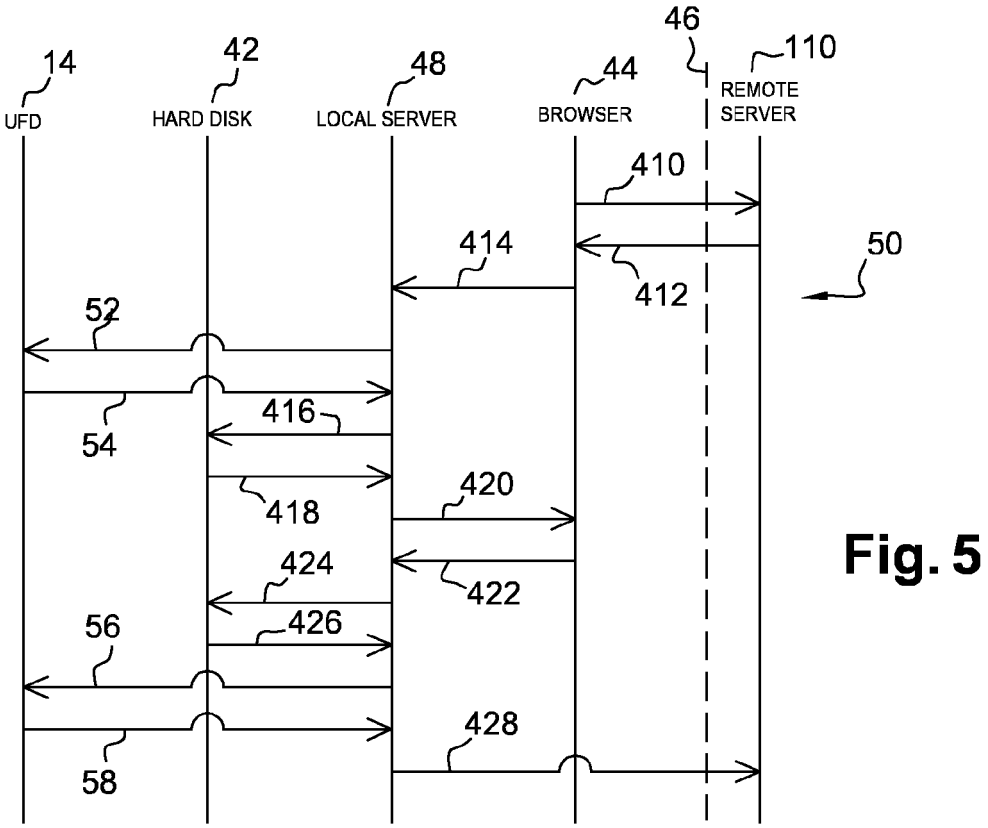
46
14
42
48
44
110
UFD
HARD DISK
LOCAL SERVER
BROWSER
REMOTE SERVER
410
414
50
412
52
54
416
418
420
424
422
56
426
428
58

Fig. 5

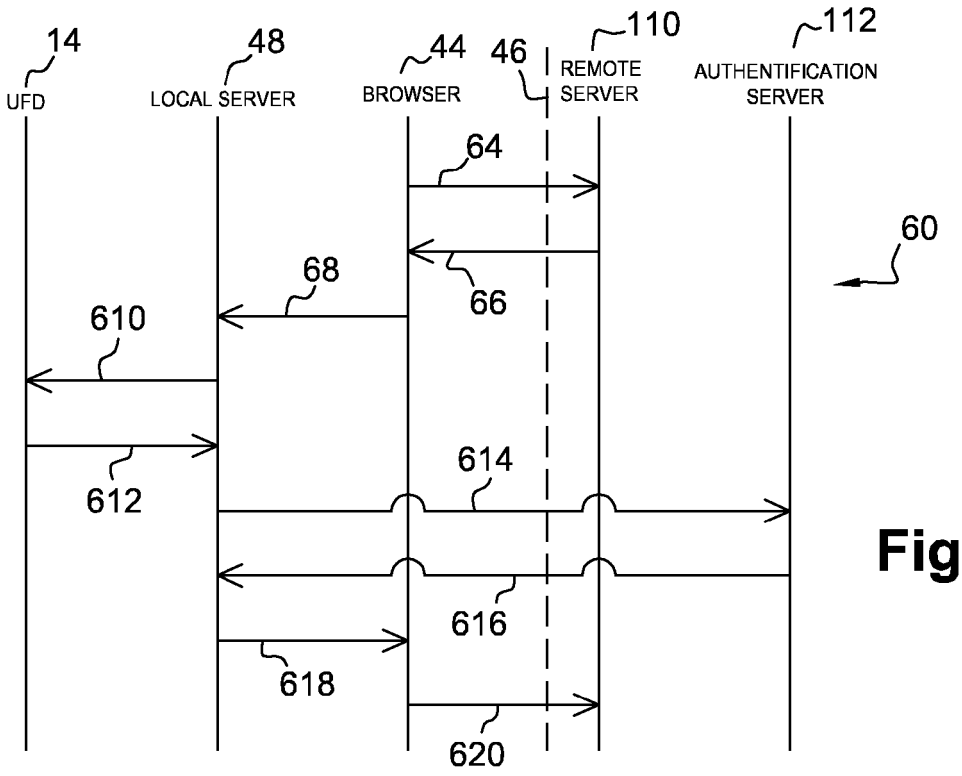
14
UFD
48
LOCAL SERVER
44
BROWSER
46
110
REMOTE SERVER
112
AUTHENTIFICATION SERVER
64
66
68
610
612
614
616
618
620
60

Fig. 6

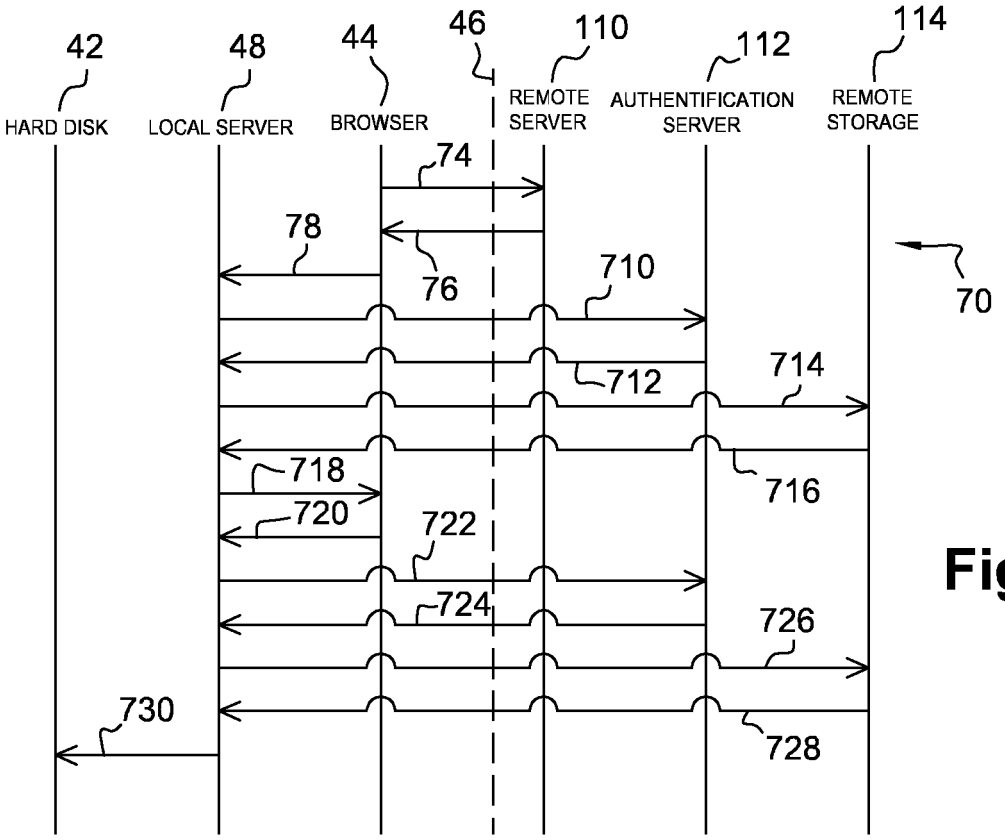
42
HARD DISK
48
LOCAL SERVER
44
BROWSER
46
110
REMOTE SERVER
112
AUTHENTIFICATION SERVER
114
REMOTE STORAGE
74
76
78
710
712
714
716
718
720
722
724
726
728
730
70

Fig. 7

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 08 30 0234

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	US 2004/128347 A1 (MASON JEFFREY [US] ET AL) 1 July 2004 (2004-07-01) * paragraphs [0008], [0009] * -----	1-15	INV. H04L29/08 G06F17/30 H04L29/06
A	US 6 167 453 A (BECKER CRAIG HENRY [US] ET AL) 26 December 2000 (2000-12-26) * column 2, line 14 - column 3, line 12; claim 1 * -----	1-15	H04L12/22

TECHNICAL FIELDS SEARCHED (IPC)

H04L
G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
The Hague	15 January 2009	Veen, Gerardus

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

1

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 08 30 0234

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

15-01-2009

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004128347 A1	01-07-2004	AU 2003291175 A1	29-07-2004
		WO 2004061567 A2	22-07-2004
US 6167453 A	26-12-2000	NONE	

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82